



面向对象的软件构造实践

实验五
(2学时)

2024春



HITSZ 实验与创新实践教育中心
Education Center of Experiments and Innovations, HITSZ

用户界面

事件处理

图形系统

数据存储
与展示

音乐音效

网络编程

模块功能：实现音乐音效控制

① 实现各种音乐音效的播放和停止



- 掌握使用MediaPlayer和SoundPool播放音频的方法。

5.1 音频播放

5.2 MediaPlayer

5.3 SoundPool

- Android的音频文件存放在应用项目的res/raw目录下。
- Android平台中播放音频有两种方式：

MediaPlayer:

MediaPlayer类会创建工作线程，进行异步操作，不会阻塞UI线程，适合比较长且对时间要求不高的情况（例如播放后台音乐、歌曲等）。

SoundPool:

SoundPool类直接将音乐加载到内存中，由当前线程播放，适合短促且对反应速度比较高的情况（例如播放游戏音效或按键声等）。

5.2 MediaPlayer

- 准备音频资源文件;
- 调用MediaPlayer.create(Context context, int resid)方法创建MediaPlayer对象, 同时通过资源ID直接加载音频文件;

```
bgMP = MediaPlayer.create(context, R.raw.bgm);
```

- 调用start()播放音频, 调用setLooping设置循环播放;

```
bgMP.start();  
bgMP.setLooping(true);
```

- 调用pause()暂停播放, 如果从暂停处继续播放;

```
position = bgMP.getCurrentPosition();  
bgMP.seekTo(position);  
bgMP.start();
```

- 停止播放并释放MediaPlayer

```
bgMP.stop();  
bgMP.release();  
bgMP = null;
```

创建对象

使用**SoundPool.Builder**来创建SoundPool对象。

```
SoundPool mysp;  
AudioAttributes audioAttributes = null;  
audioAttributes = new AudioAttributes.Builder()  
    .setUsage(AudioAttributes.USAGE_MEDIA)  
    .setContentType(AudioAttributes.CONTENT_TYPE_MUSIC)  
    .build();  
mysp = new SoundPool.Builder()  
    .setMaxStreams(10)  
    .setAudioAttributes(audioAttributes)  
    .build();
```

- setMaxStreams: 用于设置播放的流的最大数量。
- setAudioAttributes: 用于设置音频媒体的属性集。

加载音频资源

SoundPool可以通过load()方法来加载一个音频资源。

```
soundPoolMap = new HashMap<Integer,Integer>();  
soundPoolMap.put(1,mysp.load(context: this,R.raw.dingdong, priority: 1));  
soundPoolMap.put(2,mysp.load(context: this,R.raw.dingling, priority: 1));  
soundPoolMap.put(3,mysp.load(context: this,R.raw.guzhang, priority: 1));  
soundPoolMap.put(4,mysp.load(context: this,R.raw.background, priority: 1));
```

一个SoundPool对象能同时加载多个音频，所以可以通过多次调用load函数来加载。每次加载成功将返回一个非0的声音soundID，之后在播放时根据这个ID找到对应音频。

播放控制

由于一个SoundPool对象可以加载多个音频，所以在播放控制时需要指定soundID号。

```
myp.play(soundPoolMap.get(1), leftVolume: 1, rightVolume: 1, priority: 0, loop: 0, rate: 1.2f);
```

- soundID, 指定播放音频的ID号, 该ID由load()方法返回。
- leftVolume, 左声道的音量。
- rightVolume, 右声道的音量。
- priority, 流的优先级, 值越大优先级高。当同时播放数量超出了最大支持数时, 这个优先级会决定SoundPool对该流的处理。
- loop, 循环播放的次数, -1表示无限循环, 0表示不循环, 其他值为播放loop+1次。
- rate, 播放的速率, 范围为0.5-2.0。0.5为减慢一半速率, 1.0为正常速率, 2.0为加快两倍速率。

5-1 使用MediaPlayer播放背景音乐

- 若音乐开关打开，进入游戏主界面后，使用MediaPlayer循环播放背景音乐；
- 若音乐开关打开，boss机出现时**暂停**背景音乐，播放boss机音乐，boss机消失后停止boss机音乐，**继续播放**背景音乐；
- 若音乐开关打开，游戏结束时，停止背景音乐和boss机音乐的播放。
- 若音乐开关关闭，不播放任何音乐。

5-2 使用SoundPool播放音效

- 若音乐开关打开，游戏过程中，使用SoundPool播放子弹、道具音效；
- 若音乐开关打开，游戏结束时，播放游戏结束音效。
- 若音乐开关关闭，不播放任何音效。

5-1 使用MediaPlayer

1. 在res下创建raw文件夹，将音乐资源放到res/raw文件夹下；
2. 创建package，命名为music；
3. 在music文件夹下创建Java类MyMediaPlayer，此类实现使用MediaPlayer播放音乐的代码；
4. 在BaseGame中调用MyMediaPlayer类中的方法控制音乐播放，需要播放的音乐：
 - bgm.wav：游戏背景音乐
 - bgm_boss.wav: boss机背景音乐

注：boss机背景音乐出现时，要暂停游戏背景音乐。

5-2 使用SoundPool

1. 在music文件夹下创建Java类MySoundPool，此类实现使用SoundPool播放音效的代码；
2. 在BaseGame中调用MySoundPool类中的方法控制音效播放。需要播放的音效有：
 - bullet_hit.wav: 敌机子弹击中英雄机或者英雄机子弹击中敌机时播放；
 - bomb_explosion.wav: 英雄机和炸弹道具碰撞时播放（**可选**）；
 - get_supply.wav: 英雄机和非炸弹道具碰撞时播放；
 - game_over.wav: 游戏结束时播放。

中期检查标准

项目	分数	检查点	分数
实验1	20分	• 所有团队成员均能在模拟器中成功运行项目	20分
实验2	20分	• 游戏首页有一个“开始游戏”按钮和一组音乐开关，默认关闭音乐	10分
		• 单机难度选择界面有三个难度选择按钮	5分
		• 点击开始游戏，从游戏首页跳转到单机难度选择界面	5分
实验3	20分	• 根据游戏难度的不同，进入不同的游戏界面	8分
		• 实现飞机大战游戏的基本功能，生成敌机、子弹发射、碰撞检测、道具生成等	6分
		• 在左上角实时显示英雄机的得分和血量	6分

中期检查标准

项目	分数	检查点	分数
实验4	20分	• 正确显示当前难度	4分
		• 显示当前难度对应的排行榜数据，包括排名，姓名，得分，时间	6分
		• 可以删除排行榜数据，有删除提示，删除后正确更新视图显示	4分
		• 能够正确保存当前排行榜数据，保存成功有提示	6分
实验5	20分	• 若音乐开关打开，游戏开始时播放背景音乐	4分
		• 若音乐开关关闭，游戏时不播放任何音频	4分
		• 若音乐开关打开，boss机出现时播放boss机音乐，背景音乐暂停，boss机消失后背景音乐继续，boss机音乐停止	4分
		• 若音乐开关打开，游戏结束时停止播放所有音频	4分
		• 若音乐开关打开，游戏时正确播放各种音效	4分
总计	100分		100分